Atty. Docket No.: 042390.P6729
Express Mail No.: EL466333716US

APPLICATION FOR UNITED STATES PATENT

FOR

SYSTEM AND METHOD FOR REFRESHING IMAGING DEVICES OR DISPLAYS ON A PAGE-LEVEL BASIS

Inventor(s):

SCOTT A. ROSENBERG
RICHARD W. JENSEN

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard, Seventh Floor
Los Angeles, California 90025-1026
(714) 557-3800

BACKGROUND

1. Field

This invention generally relates to the field of cathode ray tubes (CRTs).

2. Background

Traditional display systems target a cathode ray tube (CRT) as their final imaging device. A CRT is typically updated in a raster fashion and require frequent refresh of the image being displayed in order to avoid perceived flickering by the user. Updating and refreshing the CRT in such manner is highly inefficient.

A new class of non-raster based imaging devices, including but not limited to liquid crystal displays (LCD), currently exists. These non-raster based imaging devices are typically "active matrix" devices, where pixels on the devices can be individual accessed and modified through the use of one or more switches at each pixel. The individual accessibility of pixels on these non-raster based imaging devices allows the pixels to be randomly turned on or off in a non-raster fashion. However, this updating and refreshing technique is inefficient as well.

BRIEF DESCRIPTION OF THE DRAWINGS

Figures 1A and 1B show exemplary systems in accordance with the current invention.

Figure 2 shows an exemplary image frame.

Figures 3A and 3B illustrate embodiments of memory configurations representing an image frame.

Figure 4 illustrates the concept of temporal coherence.

Figure 5 is a flow diagram outlining the process of performing a drawing operation to fully or partially generate an image.

Figure 6 is a flow chart outlining the process employed to refresh or update the imaging device or display.

DESCRIPTION

The present invention provides a system and method for refreshing imaging devices or displays on a page-level basis.

Figure 1A shows an exemplary system in accordance with the current invention. The "system" includes, but is not limited or restricted to a computer (e.g., desktop, laptop, hand-held, etc.). The system 100 includes a bus 105 coupling together general purpose microprocessor 110, graphics processor 115, display controller 120, and memory controller 125. It should be noted that the system 100 can also include multiple graphics processors $115_1 \ldots 115_N$ as shown in Figure 1B, where "N" is a positive integer. Memory controller 125 is operatively coupled to memory 130 to control read and write accesses to memory 130. Display controller 120 is operatively coupled to image device or display 135 to control read and write accesses to imaging device or display 135.

The drawing of images or visual information can be performed by general purpose microprocessor 110, by graphics processor(s) 115, or by a combination of general purpose microprocessor 110 and graphics processor(s) 115. Representations of images or visual information are typically deposited into image frames stored in memory 130. As will be described later, memory 130 is divided into memory pages in support of well-known memory paging schemes. Display controller 120 periodically reads the image frames stored in memory 130 and sends these image frames to imaging device or display 135 for presentation.

Figure 2 shows an exemplary image frame 200. The image frame 200 is typically divided into tiles $205_{0,0} \ldots 205_{X,Y}$, where "X" and "Y" are positive integers. Each tile represents $205_{0,0} \ldots 205_{X,Y}$ a two-dimensional region of pixels of the image frame. Images 210, 215, 220 can span over multiple tiles, as shown in Figure 2. However, images can also be contained within a tile. In accordance with the present invention and as discussed

below, the content of each tile $205_{0,0}...205_{X,Y}$ is deposited in one memory page to take advantage of the spatial coherence generally demonstrated by drawing operations to improve the drawing speed. "Spatial coherence" refers to the concept that a drawing operation is more likely to create or modify a pixel that is close to the last pixel that was created or modified than to create or modify a randomly chosen pixel.

Figure 3A illustrates one embodiment of a memory configuration representing an image frame 300. The illustrated memory configuration is referred to as the "Packed-RGB" configuration. As stated above and illustrated in Figure 2, each image frame is divided into tiles. The content of each tile is stored in a memory page 310_1, 310_2,..., 310_M, where "M" is a positive integer. In the Packed-RGB configuration, RGB-color components $305_{0,0}$, $305_{0,1}$ of one pixel are deposited or packed together in contiguous location in memory. Furthermore, color components of contiguous pixels of a tile are deposited or packed contiguously. For example, color components $305_{0,0}$ of the pixel located at coordinate (0,1) of a tile can be stored in memory next to color components $305_{0,1}$ of the pixel located at coordinate (0,0) of the same tile. In addition, color components of pixels located within one tile of the image frame are stored within the same memory page.

Figure 3B illustrates an alternative embodiment of a memory configuration representing an image frame 300. The illustrated memory configuration is referred to as the "Multi-Plane" configuration. In the Multi-Plane configuration, the content of each image frame 300 is deposited in three color planes, including (1) red plane (R-plane) 315, (2) green plane (G-plane) 320, and (3) blue plane (B-plane) 325. RGB-color components of pixels are separated and deposited in corresponding color planes. Accordingly, red (R) components 330 are deposited in the R-plane 325; green (G) components 335 are deposited in the G-plane 320; and blue (B) components 340 are deposited in the B-plane 315.

Each color plane 315, 320, 325 includes multiple memory pages. As stated above and illustrated in Figures 2, each image frame is divided into tiles. The content of each tile is stored in a memory page. Furthermore, color components of contiguous pixels are deposited or packed contiguously in the appropriate color plane. In addition, color components of pixels located within one tile of the image frame are stored within the same memory page in the appropriate color plane.

In one embodiment, memory pages having a size of 4-Kilobyte (Kbyte) is employed. In this embodiment, each 4-Kbyte memory page can hold the content of tiles having a dimension of 64 pixels by 64 pixels. In this embodiment, accesses within a tile of 64 pixels by 64 pixels falls accordingly within the same memory page. It should be noted, however, that memory pages having sizes other than 4-Kbyte can be used.

As stated above and shown in Figures 1A and 1B, the drawing of images can be performed by general purpose microprocessor 110, by graphics processor(s) 115, or by the combination of microprocessor 110 and the graphics processor(s) 115. Representations of images or visual information are generated and deposited into image frames. Each image frame is divided into tiles. The content of each tile is stored in one memory page. Display controller 120 periodically reads the image frames and sends these image frames to the display or imaging device for presentation. Display controller 120 sends these image frames to the display one memory page at a time for efficiency purposes.

In most image applications, temporal coherence occurs. Temporal coherence refers to the concept that over some period of time, the content of a majority of the tiles of image frames generated consecutively over time would typical remain the same. Figure 4 illustrates the concept of temporal coherence. For example, tile (0,0) 405_1, 405_2, 405_3 remains unchanged from the first image frame 400_1, to the second image frame 400_2, and to the third image frame 400_3.

Accordingly, to improve the efficiency of the process of updating or refreshing the display or imaging device, display controller 120 (shown in Figures 1A and 1B) employs a process where only modified pages are sent to the imaging device for representation.

Figure 5 is a flow chart outlining the process of performing a drawing operation. In block 510, images or visual information are generated, and the content of image frames used to store those generated images are updated. In block 515, memory pages corresponding to the tiles that have been updated due to the generation of the image or visual information are marked as being "modified" or "dirty".

Figure 6 is a flow chart outlining the process employed to refresh or update the imaging device or display with only memory pages that have been modified, known as "dirty" memory pages. In block 610, the current memory page is initialized to be the first memory page of the image frame. In block 615, if the current memory page has been marked as "modified" by a drawing operation, as shown in Figure 5 and described in the accompanying text, the current memory page is sent to the display or imaging device to be presented (block 620). The current memory page is then marked as "unmodified" (block 625). If the current memory page has not been marked as "modified", the memory page is sent to the display or imaging device only if the display or image device requires an update or refresh (block 630). In block 630, a query is performed to determine whether the last memory page of the image frame has been processed. If the last memory page of the image frame has not been processed, the current memory page is set equals to the next memory page in the image frame (block 635). The sequence of actions in blocks 615 to 625 are then repeated. If the last memory page of the image frame has been processed, the process of refreshing or updating the display or imaging device is then completed.

It should be noted that the functional components illustrated in Figures 1A and 1B and discussed above may be implemented in hardware or software. If the aforementioned functional components are implemented as a software program, the functionality of these components can be emulated by one or more sub-programs, which can be stored on a system-readable medium, such as floppy disk, hard drive, CD-ROM, digital video disk, tape, memory, or any storage device that is accessible by the system.

While certain exemplary embodiments have been described and shown in the accompanying drawings, it is to be understood that such embodiments are merely illustrative of and not restrictive on the broad invention, and that this invention not be limited to the specific constructions and arrangements shown and described, since various other modifications may occur to those ordinarily skilled in the art.